SPEECH CODING AT 4800 BPS FOR MOBILE SATELLITE COMMUNICATIONS†

Allen Gersho, Wai-Yip Chan, Grant Davidson‡, Juin-Hwey Chen‡, and Mei Yong

Communications Research Laboratory
Department of Electrical & Computer Engineering
University of California, Santa Barbara, CA 93106

ABSTRACT

At UCSB, a speech compression project has recently been completed for the NASA Mobile Satellite Experiment (MSAT-X) to develop a speech coding algorithm suitable for operation in a mobile satellite environment aimed at providing telephone quality natural speech at 4.8 kbps. The work has resulted in two alternative techniques which achieve reasonably good communications quality at 4.8 kbps while tolerating vehicle noise and rather severe channel impairments. The algorithms are embodied in a compact self-contained prototype consisting of 2 AT&T 32-bit floating-point DSP32 digital signal processors (DSP). A Motorola 68HC11 microcomputer chip serves as the board controller and interface handler. On a wirewrapped card, the prototype's circuit footprint amounts to only 200 sq cm, and consumes about 9 watts of power.

1. Introduction

At the University of California at Santa Barbara (UCSB), we have completed a three year project for the NASA Mobile Satellite Experiment (MSAT-X) managed by the Jet Propulsion Laboratory (JPL). The objective of the project is the exploration of novel speech compression algorithms, culminating in a prototype voice codec that delivers "near-toll" quality speech at a data rate of 4800 bps and is suitable for use in MSAT-X field trials. An additional goal is to exhibit the potential for low fabrication cost, and low space and power consumption. Resulting from this work are two coding algorithms, VAPC (Vector Adaptive Predictive Coding) [1] and PVXC (Pulse Vector Excitation Coding) [2], with complexity in terms of simulation Flops count of 3.5 and 1.5 MFlops respectively, where one Flops stands for one floating-point multiply-and-add per second and a hardware prototype that implements both algorithms. At this stage, the prototypes (Fig. 1) are slated to enter field trials in the summer of 1988. We believe that, coupled with up-to-date product engineering, the prototype can be readily transformed into a cost effective voice coding component for mobile satellite terminals.

2. System Overview

2.1. Algorithm Review

The theme of our algorithm development was the extensive use of vector quantization (VQ), a powerful generic technique for efficient coding of sets of parameters that characterize attributes of a speech sound. With VQ, a relatively short binary word is often sufficient for accurately specifying the amplitude of a large number of parameter values or waveform samples needed for reproducing speech sounds at the receiver. The two algorithms we have developed evolved from two different approaches. VAPC stems from a vector quantized and

† This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration.

‡ G. Davidson is currently with Dolby Laboratories. J.-H. Chen is currently with Codex Corporation.

frame-adaptive extension of DPCM and PVXC from excitation coding techniques, such as Multipulse LPC and Code Excited Linear Prediction (CELP). In VAPC, a frame of speech samples is analyzed to determine the pitch, pitch predictor, LPC predictor, and gain. The speech frame is partitioned into vectors, from each of which the pitch prediction is removed. Then a frame-specific *zero-state-response* (ZSR) codebook is synthesized from a chain consisting of a gain-normalized residual codebook, the quantized gain and LPC predictor, plus a perceptual weighting filter derived from the quantized predictor. By searching for the best matching codevectors in the ZSR codebook, the prediction residual of the frame is efficiently vector-quantized. At the decoder, coding noise in the resynthesized speech is suppressed by an adaptive postfilter that includes compensation for spectral tilt.

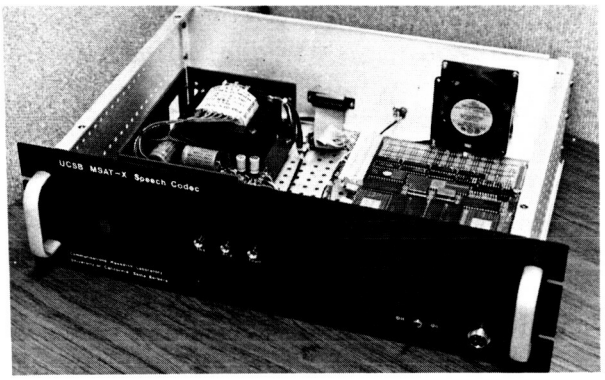

Figure 1. Codec Prototype

The analysis-by-synthesis model of PVXC consists of a codebook of excitations, a gain scaling operation, pitch and short-term LPC inverse filters, and a perceptual weighting filter, in that order. Quantized prediction parameters are employed in both synthesis filters, whereas unquantized LPC coefficients are used to obtain the parameters of the perceptual weighting filter. Per-frame short-term LPC analysis is performed prior to pitch extraction and long-term prediction coefficient computation. The excitation codebook driving the synthesis chain contains sparse pulse-excitation vectors, for each of which a quantity related to the energy of its synthetic speech vector is precomputed in every frame period. A separate gain parameter is associated with each best-matching (in the synthetic speech space) excitation vector selected from the codebook. In the decoder, a postfilter adapted by the received short-term LPC parameters enhances the decoded speech.

Both algorithms encode telephony bandwidth speech sampled at 8 KHz. Speech is analyzed without frame overlap at 20 ms and 22.5 ms intervals, respectively for PVXC and VAPC; the corresponding data frame sizes are 96 and 108 bits. Both algorithms employ VQ to code the residual or excitation signals. The vector dimension for VAPC is 20 samples, or 2.5 ms, and is 40 samples or 5 ms for PVXC. The LPC quantization schemes for both coders are quite similar, although the quantization tables and bit allocations differ. Ten LPC prediction coefficients are converted to Line Spectral Pair (LSP) parameters, processed by Switched-Adaptive Interframe Vector Prediction (SIVP) [2], and then scalar quantized. VQ is also applied to quantizing the 3rd order pitch predictor in each algorithm. The remaining parameters are scalar quantized: pitch; and gains, one for each frame in VAPC and each vector in PVXC. The VQ codebooks are trained from a multiple-speaker speech database. Unstructured, the codebooks can be partitioned among parallel processing element to achieve higher throughput. Decoder processing is relatively non-intensive, with VQ table lookups replacing the intensive VQ searches in the encoder, and includes such communication tasks as frame

alignment and error control.

2.2. Channel and Operation Conditions

In the mobile environment, we are confronted with rather severe conditions at the acoustic frontend and in the transmission channel. The background noise in a roaming mobile could be very loud, nonstationary, and have periodic components. Simulation and implementation-based tests have confirmed that, in the presence of additive noise, the two schemes have the robustness of higher bit-rate waveform coders[3] but not the serious degradations characteristic of low-rate vocoders. JPL simulation studies found an average bit-error rate of 10^{-3} in the MSAT-X channel and, in spite of the countering effect of extensive interleaving in the modem, the errors are predominately bursty due to fading. In listening tests using error patterns derived from an MSAT-X channel simulator, significantly audible clicks were noticed only occasionally in the absence of error protection. Maintenance of frame synchronization is however critical to the overall voice quality, so part of the 4800 bps capacity must be devoted to framing. A one way end-to-end coding delay objective of 50 ms. for the codec (for an error-free, zero-delay channel) is a small fraction of the overall transmission delay of the MSAT-X channel which can include two satellite-hops of delay and other delays in the modem and the network center. Due to additional buffering for channel errors and data management, the delay in our current implementation is slightly higher, approximately 60 ms. Coding delay is minimized by transmitting coder data as close as possible to the order in which they are generated.

In a complete MSAT-X Voice/Data Terminal, the speech codec is located between a handset and the JPL *Terminal Processor* (TP) driving the modem looking into the network. The only dialing function that the codec is responsible for is hook switch monitoring; an "off-hook" signal activates the coder. Voice data is transferred synchronously at 4800 bps to the TP; the data clock originates from the same source as the local sampling clock. From the TP, the codec receives Receive voice data (clocked independently of the Transmit stream), reception condition information, and all the Transmit and Receive clocks. Frame timing, though transparent to the transport network, is indicated to the TP so that it can enforce frame alignment.

2.3. Framing and Error Control

The low channel bandwidth limits the number of bits available for link maintenance. We have therefore chosen to realize only 4 overhead bits/frame (or 200 bps), one framing bit and 3 error control bits. The number of framing bits needed per frame is determined by the required reframing time, which is usually desired to be inversely related to the fading dropout rate of the channel. For random bit patterns and an error-free channel, the mean reframing times using 1 framing bit is about 200 ms, assuming a false locking probability of 10^{-3}. To prevent false locking onto imitative patterns in the data stream, and to simplify the frame-tracking firmware, we have opted to use the simple periodic framing pattern 1010... In acquisition mode, a frame-size register that keeps track of the surviving framing-bit candidates is logical-ANDed with the exclusive-OR of consecutive frame-size blocks of Receive data; this is equivalent to correlation matching with 1 correlation memory bit per data bit. Since loss of frame synchronization occurs rarely, and since the preponderant burst errors tend to garble some sync bits whenever they occur, out-of-frame condition is not declared unless either a predetermined number of pattern violations in a window of frames occurs, or the modem has been asserting channel fade condition for more than a specific timeout period. During fades, the frame clock is flywheeled.

During frame synchronized epoches, some form of error control is desirable. With only 3 bits allocated for this purpose, we could only cater to the most critical need: detection of

burst error. By using each bit to check the parity of every other 3 bits in a frame, the probability of missing a block error is reduced to one-eighth. Though this is hardly a small number, our coders are quite resilient to hits even with no error control. When the parity bits do not check, a previous frame is repeated if the number of consecutive prior repetitions is below a threshold; otherwise silence is filled in. An ongoing effort in our laboratory is studying an efficient countermeasure for VQ based coders against channel errors, that reduces the degradation in speech quality without sacrificing channel bits: a generalized Gray coding technique for assigning binary indices to codevectors and codebook design accounting for channel errors[3, 4].

3. Hardware

3.1. The Signal Processor

We elected to use the commercially available AT&T-DSP32 DSP [5] primarily because of its 32-bit floating point arithmetic. This feature eliminates the need for scaling and double precision arithmetic and substantially reduces the software development time. The C-like constructs of the DSP32 assembly language also facilitated programming immensely, though this was somewhat offset by semantic constraints imposed by non-transparent pipeline latency effects. It is difficult for the DSP32 to serve additionally as a codec controller because the chip is limited in I/O support and booting from slower EPROMs is complicated by the lack of wait state generation. The instruction cycle time of the DSP32 chips used in our codec is 160 ns (6.25 MFlops).

3.2. Circuit Board

To ease development effort and to facilitate experimentation with high performance versions, our initial prototype was designed for 3 DSPs [6]. Recent studies have shown that the codec can be adapted to handle bit rates ranging as high as 16 kbps and performance improves smoothly with the bit rate. In the sequel, we will describe only the final version of our prototype, which has only two DSPs and has enough processing power to handle all bit rates of interest.

Although a single DSP possesses almost enough power to handle both encoding and decoding in real time at 4800 bps, to minimize coding delay and to facilitate the handling of the 2 asynchronous voice streams, it is desirable to run the coder and decoder on separate processors. The board was designed with sufficient built-in flexibility for servicing the voice and TP interfaces. Only the RAM chips and master clock oscillator need to be changed to accomodate faster processor chips.

Each DSP (Fig. 2) can access 8K words (32K bytes) of dedicated external RAM, plus 1K words of on-chip refreshed RAM. The DSPs' serial ports are connected to an AT&T high precision codec/filter chip and their parallel ports to the 68HC11 controller. Data transfer between DSP memories and I/O ports is via DMA. The encoder/decoder DSP receives/transmits 15-bit PCM samples from/to the codec chip. The codec chip is not unlike ordinary mu-law codec chips except that the dynamic range and SNR have been improved to 80 dB and 60 dB respectively. The chip is driven by a 2.048 MHz clock provided by the modem's modulator and this clock also is the source of the 4.8 kHz Transmit data clock. The 2.048 MHz clock is divided down to 8 kHz for A/D and D/A sampling and for codec-chip/DSP data transfer. Because the Transmit and Receive data clocks are asynchronous, the decoder has to compensate for clock frequency offset between the remote and local clocks by monitoring its incoming data buffer and D/A speech buffer for sample dropping or insertion; the absolute stability of the oscillator limits this drift to less than 1 Hz. This simple scheme allows the use of a single codec/filter

chip to perform the A/D and D/A conversion. The analog side of the codec/filter chip is connected to simple analog circuitry for side-tone and signaling tone insertion and for interfacing to a Shure handset which has low sensitivity to breath noise.

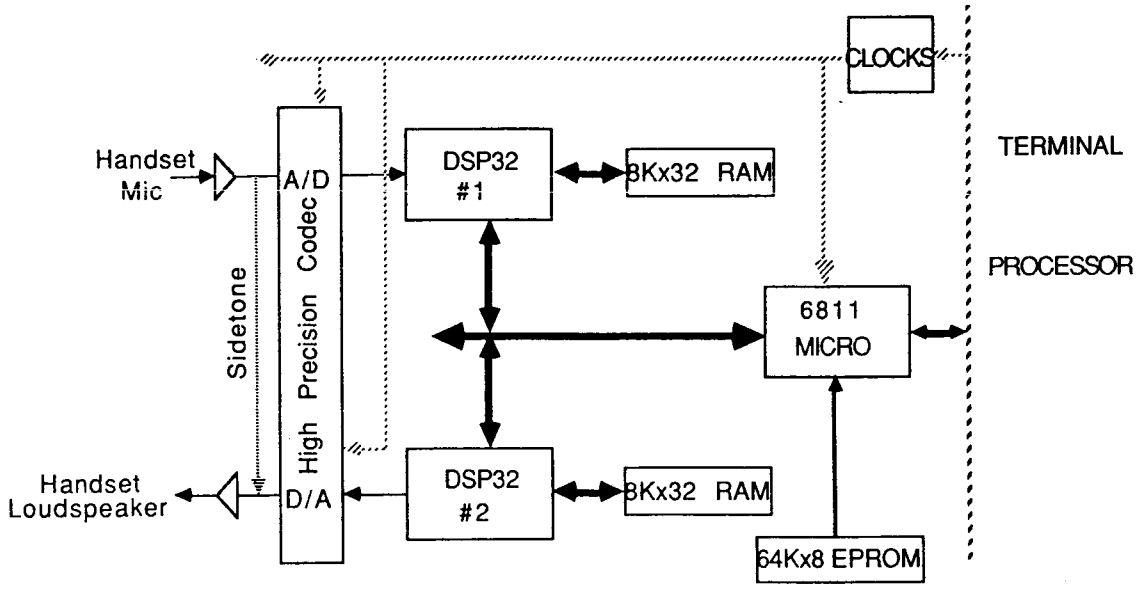

Figure 2. Prototype Board Block Diagram

The parallel ports of the DSPs are connected to an 8-bit bus mastered by a Motorola 68HC11 microcomputer chip, basically a 0.5 Mips engine. On power up, the micro copies DSP code from a 64Kx8 EPROM into the RAMs of the DSPs. Rudimentary system integrity is verified through check-summing the EPROM and comparing downloaded programs with EPROM content. With the aid of several glue chips, including PALs, the micro orchestrates the operation of the entire codec: full-duplex data transfer management across the TP Interface and the DSP/controller interfaces as well as hook monitoring. The controller can handle a real-time load up to at least 16 kbps. The health of the DSPs is monitored during run time by enabling interrupts caused by the detection of instruction parity error and data address alignment error.

The prototype codec circuitry consumes about 200 cm^2 of space and 9 watts of power on a wirewrap card. A single-DSP implementation on a printed circuit board would shrink the space and power consumption by 40%. Manufacturing cost for such a card would only be a few hundred dollars. Owing to the high precision codec and premium quality handset, the speech quality in PCM-loopback configuration is considerably better than typical digital sets.

4. Firmware

When a breakdown of the time and storage utilization of the encoder processor was examined, it became apparent that VQ precomputation and codebook searching are the dominant activities, consuming more than 60% of DSP execution time and 50% of storage space. In contrast, LPC analysis, a significant task for DSPs not so long ago, consumes only a few percent of processor time. Even SIVP quantization of the spectral envelopes requires twice as much time as LPC analysis. The dominant data structure in VAPC is its residual codebook, which holds 128 vectors, each consisting of 20 mu-law coded samples. The PVXC codebook is considerably simpler since the 256 40-dimensional vectors have only on the average 4 pulses per vector; each pulse is coded by a mu-law amplitude and a position byte.

The final processor flop counts are 3.3 times and 1.8 times the algorithmic (implementation independent) flop counts of PVXC and VAPC, respectively; these numbers give a measure of matching between the algorithms and the DSP architecture.

5. Conclusion

The real-time prototype was a necessity for verifying and fine-tuning algorithm performance in the laboratory as well as providing a resource for MSAT-X field trials. It enabled observation of such performance characteristics as degree of speaker dependence, effects of acoustical noise and background speakers, and channel originated degradations. We have, for instance, noticed a mild degree of quality variation across speakers and genders. We have also verified that extraneous signals do not cause the speech coding to breakdown as is typical of low rate vocoders.

We have described a real-time realization of two 4800 bps voice coding algorithms that exhibits cost-effective product prospect. The voice quality is quite good for the 4.8 kbps bit rate, though more effort is needed to assess the level of user acceptance amidst the selected pool of potential MSAT users. In the immediate future, we will be able to acquire performance data under realistic transmission scenarios.

Though developed for MSAT-X, the algorithms are suitable for other 4.8 kbs applications where "communication quality" is acceptable. Furthermore, substantially improved quality is attainable as the bit rate is increased and we see a strong potential for application of both algorithms for other telecommunications applications at bit-rates of 8, 9.6, and 16 kbps.

In summary, the prototype offers a reasonably satisfactory, natural voice quality and demonstrates that a compact low cost implementation of the codec is feasible with current technology.

ACKNOWLEDGMENT

UCSB graduate students who have contributed at various stages to this work are Mark Grosen, Vijaykumar Narayanan, Shihua Wang, Jae-Hsin Yao, Mei Yong, and Kenneth Zeger. We also direct our gratitude to NASA and JPL for their sponsorship of this project. We acknowledge the substantial support and cooperation of many JPL engineers and managers, and in particular, Steve Townes (currently with MITRE), William Rafferty, and Thomas Jedrey, who have served as technical liaisons with UCSB.

References

1. J.H. Chen and A. Gersho, "Real-Time Vector APC Speech Coding at 4800 bps with Adaptive Postfiltering," *Proc. ICASSP*, p. 51.3, April 1987.
2. G. Davidson, M. Yong, and A. Gersho, "Real-Time Vector Excitation Coding of Speech at 4800 bps," *Proc. ICASSP*, p. 51.4, April 1987.
3. J.H. Chen, G. Davidson, A. Gersho, and K. Zeger, "Speech Coding for the Mobile Satellite Experiment," *Proc. ICC*, June 1987.
4. K.A. Zeger and A. Gersho, "Zero Redundancy Channel Coding in Vector Quantisation," *Electronics Letter*, p. 654, May 7, 1987.
5. "WE DSP32 Digital Signal Processor Information Manual," *ATT*, June 1986.
6. W.Y. Chan, G. Davidson, J.H. Chen, and A. Gersho, "A Prototype 4800 bps Voice Terminal for the Mobile Satellite Experiment," *Proc. GLOBECOM*, Nov. 1987.